

CLAIMS

What is claimed is:

1. A method for repetitively executing a plurality of software packages at one or
2 more rates, utilizing a common set of computational resources, the method comprising the steps:
3 generating a sequence of time intervals for each of the plurality of software
4 packages, the time intervals belonging to one software package not overlapping the time intervals
5 belonging to any other of the plurality of software packages;
6 executing a plurality of software packages, each software package being executed
7 during the time intervals of its sequence of time intervals.
1. 2. The method of claim 1 wherein the plurality of software packages of the
2 "executing" step includes only valid software packages, the method further comprising the step:
3 utilizing one or more tests to identify the software packages that are valid.
1. 3. The method of claim 2 wherein one of the tests for validity is a one's complement
2 checksum test of a software package's program memory.
1. 4. The method of claim 2 wherein a software package is assigned its own dedicated
2 memory region, one of the tests for validity being whether the address returned for the software
3 package's initialization procedure lies within its dedicated memory region.
1. 5. The method of claim 4 wherein one of the tests is whether the address is returned
2 within a predetermined time.
1. 6. The method of claim 2 wherein a software package is assigned its own dedicated
2 memory region, the software package's dedicated memory region including a stack memory

PARTITIONED EXECUTIVE STRUCTURE FOR REAL-TIME PROGRAMS

CUREY, TAZARTE, BANNO & MARK

P573C

3 region and/or a heap memory region, one of the tests for validity being whether the stack memory
4 range and/or the heap memory range assigned during the execution of the software package's
5 initialization procedure and the various associated entry points lies within the software package's
6 dedicated memory region.

1 7. The method of claim 6 wherein one of the tests is whether the stack memory range
2 and/or the heap memory range and the various associated entry points are returned within a
3 predetermined time.

1 8. The method of claim 1 wherein a software package is assigned its own dedicated
2 memory region.

1 9. The method of claim 8 wherein the software package's dedicated memory region
2 includes a stack memory region, a software package's stack residing in the software package's
3 stack memory region.

1 10. The method of claim 1 wherein a software package includes background tasks as
2 well as foreground tasks, the background tasks being performed after the foreground tasks have
3 been completed.

1 11. The method of claim 10 wherein a background task is an infinite loop.

1 12. The method of claim 10 wherein the software package causes the power utilized
2 in executing the software package to be minimized after completion of the background tasks.

1 13. The method of claim 1 wherein a failure in the execution of a software package
2 causes information to be logged in a failure log.

1 14. The method of claim 13 wherein a failure in execution is linked to the software
2 package that caused the failure.

PARTITIONED EXECUTIVE STRUCTURE FOR REAL-TIME PROGRAMS

CUREY, TAZARTES, BANNO & MARK

P573C

1 15. The method of claim 13 wherein quality of performance in executing a software
2 package is represented by one or more performance-quality parameters, values of the one or more
3 performance-quality parameters being determined from the information logged in a failure log,
4 the execution of a software package being subject to a plurality of execution options, an
5 execution option being selected on the basis of one or more performance-quality parameter
6 values.

1 16. The method of claim 15 wherein the plurality of execution options are user
2 configurable.

1 17. The method of claim 15 wherein performance-quality parameters include the
2 number of failures and/or the rate of failures for one or more classes of failures recorded in a
3 software package's failure log.

1 18. The method of claim 1 wherein safety-critical software is placed in one or more
2 separate partitions thereby isolating the safety-critical software from non-safety-critical software.

1 19. The method of claim 1 wherein each of the plurality of software packages is
2 assigned its own memory block, a software package being enabled to read data only from zero or
3 more memory blocks associated with other software packages, the zero or more memory blocks
4 readable by a software package being either predetermined or determined during execution of the
5 software packages in accordance with a set of one or more rules.

1 20. The method of claim 1 wherein each of the plurality of software packages is
2 assigned its own memory block, a software package being enabled to write data only to zero or
3 more memory blocks associated with other software packages, the zero or more memory blocks

PARTITIONED EXECUTIVE STRUCTURE FOR REAL-TIME PROGRAMS

CUREY, TAZARTES, BANNO & MARK

P573C

4 writeable by a software package being either predetermined or determined during execution of
5 the software packages in accordance with a set of one or more rules.

1 21. The method of claim 1 wherein an executive software package enforces the
2 discipline that each software package executes only during the time intervals of its sequence of
3 time intervals, the executive software package determining when the execution of a software
4 package extends into a time interval belonging to the sequence of time intervals assigned to
5 another software package and performs a remedial action.

1 22. The method of claim 1 wherein the presence of those software packages that are
2 present is detected.

1 23. The method of claim 1 wherein one or more of the plurality of software packages
2 are independently compiled, linked, and loaded.

1 24. The method of claim 1 wherein a software package has its own stack, the software
2 package's stack being selected prior to executing the software package.

1 25. Apparatus for practicing the method of claim 1.

1 26. Apparatus for repetitively executing a plurality of software packages at a plurality
2 of rates, the apparatus comprising:

3 a means for generating a sequence of time intervals for each of the plurality of
4 software packages, the time intervals belonging to one software package not overlapping the time
5 intervals belonging to any other of the plurality of software packages;

6 a means for executing a plurality of software packages, each software package
7 being executed during the time intervals of its sequence of time intervals.

PARTITIONED EXECUTIVE STRUCTURE FOR REAL-TIME PROGRAMS

CUREY, TAZARTES, BANNO & MARK

P573C

1 27. The apparatus of claim 26 wherein the plurality of software packages executed by
2 the "executing" means includes only valid software packages, the apparatus further comprising:
3 a means for utilizing one or more tests to identify the software packages that are
4 valid.

1 28. The apparatus of claim 27 wherein one of the tests for validity is a one's
2 complement checksum test of a software package's program memory.

1 29. The apparatus of claim 27 wherein a software package is assigned its own
2 dedicated memory region, one of the tests for validity being whether the address returned for the
3 software package's initialization procedure lies within its dedicated memory region.

1 30. The apparatus of claim 29 wherein one of the tests is whether the address is
2 returned within a predetermined time.

1 31. The apparatus of claim 27 wherein a software package is assigned its own
2 dedicated memory region, the software package's dedicated memory region including a stack
3 memory region and/or a heap memory region, one of the tests for validity being whether the stack
4 memory range and/or the heap memory range assigned during the execution of the software
5 package's initialization procedure and the various associated entry points lies within the software
6 package's dedicated memory region.

1 32. The apparatus of claim 31 wherein one of the tests is whether the stack memory
2 range and/or the heap memory range and the various associated entry points are returned within
3 a predetermined time.

PARTITIONED EXECUTIVE STRUCTURE FOR REAL-TIME PROGRAMS

CUREY, TAZARTES, BANNO & MARK

P573C

1 33. The apparatus of claim 26 wherein a software package is assigned its own
2 dedicated memory region.

1 34. The apparatus of claim 33 wherein the software package's dedicated memory
2 region includes a stack memory region, a software package's stack residing in the software
3 package's stack memory region.

1 35. The apparatus of claim 26 wherein a software package includes background tasks
2 as well as foreground tasks, the background tasks being performed after the foreground tasks
3 have been completed.

1 36. The apparatus of claim 35 wherein a background task is an infinite loop.

1 37. The apparatus of claim 35 wherein the software package causes the power utilized
2 in executing the software package to be minimized after completion of the background tasks.

1 38. The apparatus of claim 26 wherein a failure in the execution of a software
2 package causes information to be logged in a failure log.

1 39. The apparatus of claim 38 wherein a failure in execution is linked to the software
2 package that caused the failure.

1 40. The apparatus of claim 38 wherein quality of performance in executing a software
2 package is represented by one or more performance-quality parameters, values of the one or more
3 performance-quality parameters being determined from the information logged in a failure log,
4 the execution of a software package being subject to a plurality of execution options, an
5 execution option being selected on the basis of one or more performance-quality parameter
6 values.

PARTITIONED EXECUTIVE STRUCTURE FOR REAL-TIME PROGRAMS

CUREY, TAZARTES, BANNO & MARK

P573C

1 41. The apparatus of claim 40 wherein the plurality of execution options are user
2 configurable.

1 42. The apparatus of claim 40 wherein performance-quality parameters include the
2 number of failures and/or the rate of failures for one or more classes of failures recorded in a
3 software package's failure log.

1 43. The apparatus of claim 26 wherein safety-critical software is placed in one or
2 more separate partitions thereby isolating the safety-critical software from non-safety-critical
3 software.

1 44. The apparatus of claim 26 wherein each of the plurality of software packages is
2 assigned its own memory block, a software package being enabled to read data only from zero or
3 more memory blocks associated with other software packages, the zero or more memory blocks
4 readable by a software package being either predetermined or determined during execution of the
5 software packages in accordance with a set of one or more rules.

1 45. The apparatus of claim 26 wherein each of the plurality of software packages is
2 assigned its own memory block, a software package being enabled to write data only to zero or
3 more memory blocks associated with other software packages, the zero or more memory blocks
4 writeable by a software package being either predetermined or determined during execution of
5 the software packages in accordance with a set of one or more rules.

1 46. The apparatus of claim 26 wherein an executive software package enforces the
2 discipline that each software package executes only during the time intervals of its sequence of
3 time intervals, the executive software package determining when the execution of a software

PARTITIONED EXECUTIVE STRUCTURE FOR REAL-TIME PROGRAMS

CUREY, TAZARTES, BANNO & MARK

P573C

4 package extends into a time interval belonging to the sequence of time intervals assigned to
5 another software package and performs a remedial action.

1 47. The apparatus of claim 26 wherein the presence of those software packages that
2 are present is detected.

1 48. The apparatus of claim 26 wherein one or more of the plurality of software
2 packages are independently compiled, linked, and loaded.

1 49. The apparatus of claim 26 wherein a software package has its own stack, the
2 software package's stack being selected prior to executing the software package